

### REMARKS

Claims 1-7 are pending in this application. Claims 1 and 5 have been amended to more distinctively claim Applicants' invention. Claims 2-4 and 6 have been cancelled.

#### Claims Rejected under 35 USC §112

Claim 3 has been cancelled.

#### Claims 1-7 are Rejected under 35 USC §102

Claims 1-7 are rejected under 35 U.S.C. §102(e) as being anticipated by Rathbun (U.S. Patent No. 6,138,123). The Examiner stated that

As to Claim 1, Rathbun teaches a method (see Abstract) to perform an add operation from a root to a leaf (see figures 4-8, and see column 6, lines 49-63, where "add operation" is read on "insert() function"), comprising the steps of: identifying a root (see column 14, lines 4-11; identifying any unused leaf (see column 26, lines 54-65, where "unused leaf" is read on "empty cell") and traversing from the root to the any unused leaf (see column 7, lines 4-22).

Claim 1 recites a method to perform an add operation from a root to a leaf, comprising the steps of: in a data structure comprising a heap, comprising: identifying a root in the data structure; identifying any unused leaf in the data structure; and traversing from the root to the any unused leaf in the data structure, the traversing step comprises the step of percolating values within the data structure from the root to any unused leaf, the traversing step comprises the step of not percolating values within the data structure from the root to any unused leaf, the percolating values comprising one or more holes.

First, the data structure in Rathbun as described in Column 7, lines 4-22 and in Figure 68 is different in which there are three separate B-trees, as indicated by the local index 1, the local index 2 and the local index 3. Moreover, the query simply travels down the rightmost local link, locating the range (t-z). (Rathbun, Col. 7, lines 4 – 22). Even so, Claim 1 is provides greater flexibility by “traversing from the root to the any unused leaf in the data structure, the traversing step comprises the step of percolating values within the data structure from the root to any unused leaf.”

Continuing in Col. 8, beginning in line 37, Rathbun states that data structures can be based on heaps, B-trees, and binary search trees, etc. As described in the background section in Applicants’ application, there is an extensive discussion on conventional heap data structures. Conversely, Claim 1 recites features that are beyond traditional heap data structures.

Claims 2-4 have been cancelled.

Claim 5 recites a method for performing a remove operation in a data structure, comprising the steps of: removing a value from a root that leaves a hole; and percolating the hole to a leaf position, wherein in the percolating step, comprising the step of percolating the hole to any arbitrary leaf position in the data structure. (Emphasis Added).

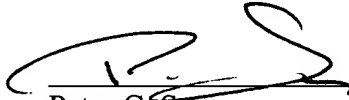
Claim 7 depends from Claim 5 and it is likewise allowable for at least the same reasons as Claim 5 which are incorporated by reference.

CONCLUSION

Claims 1, 5 and 7 are pending in this application. In view of the above, it is respectfully submitted by Applicants that the claims are in condition for allowance. If the Examiner's action is other than allowance, the Examiner is invited to telephone Applicants' attorney at the number noted below.

Respectfully submitted,

10-7-2003  
Date

  
Peter C. Su  
Attorney for Applicants  
Reg. No. 43,939

Address: **FERNANDEZ AND ASSOCIATES LLP**  
Patent Attorneys  
1047 El Camino Real  
Menlo Park, CA 94025

Phone: (650) 325-4999  
Fax: (650) 325-1203  
Email: [iploft@iploft.com](mailto:iploft@iploft.com)